Customer Shopping Analysis - Subham Choudhary – SQL Project

In this SQL project, we aim to analyze customer shopping data from a fictional retail store. The dataset includes information such as customer demographics, purchase details, payment methods, and shopping locations. Our goal is to perform various SQL operations to gain insights into customer behavior, product preferences, and transaction patterns.

1. The Data Looks like

```
INSERT INTO Invoices (customer_id, gender, age, category, quantity, price, payment_method, invoice_date, shopping_mall)
VALUES
('C241288', 'Female', 28, 'Clothing', 5, 1500.4, 'Credit Card', '2022-05-08', 'Kanyon'),
('C111565', 'Male', 21, 'Shoes', 3, 1800.51, 'Debit Card', '2021-12-12', 'Forum Istanbul'),
('C266599', 'Male', 20, 'Clothing', 1, 300.08, 'Cash', '2021-09-11', 'Metrocity'),
('C169650', 'Female', 66, 'Clothing', 2, 600.16, 'Credit Card', '2022-09-08', 'Istinye Park'),
('C337046', 'Female', 53, 'Books', 4, 60.6, 'Cash', '2021-10-24', 'Kanyon'),
('C227836', 'Female', 28, 'Clothing', 5, 1500.4, 'Credit Card', '2022-05-24', 'Forum Istanbul'),
('C121056', 'Female', 49, 'Cosmetics', 1, 40.66, 'Cash', '2022-03-13', 'Istinye Park'),
('C293112', 'Female', 32, 'Clothing', 2, 600.16, 'Credit Card', '2021-01-13', 'Mall of Istanbul'),
('C293455', 'Male', 69, 'Clothing', 3, 900.24, 'Credit Card', '2021-11-04', 'Metrocity'),
```

- Create a Table with Unique ID Numbers: We'll replace the existing invoice numbers with unique ID numbers from 1 to the total number of invoices.

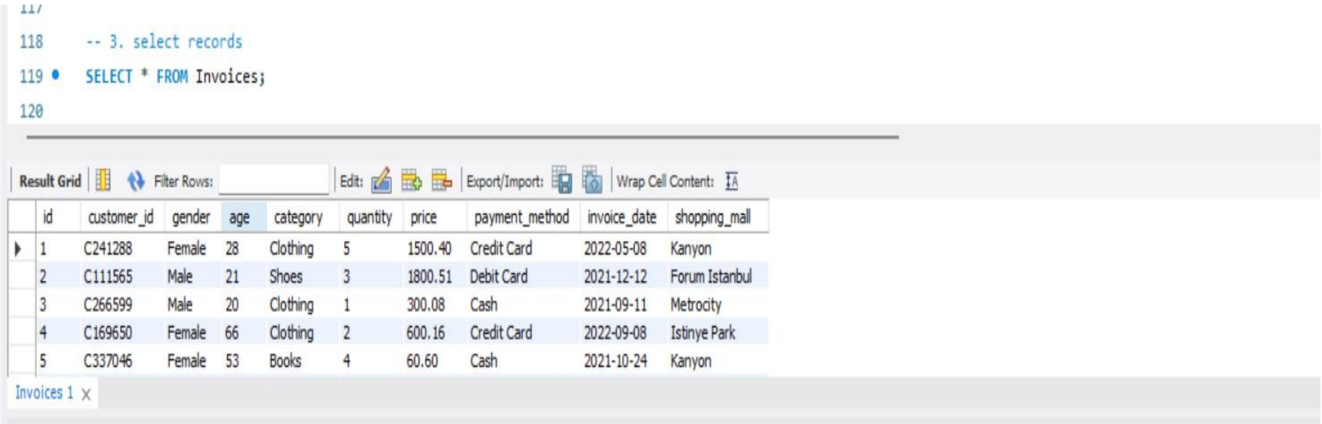
2. Insert Data into the Table: Insert the modified data into the new table.

Create a Table with Unique ID Numbers

```
CREATE TABLE Invoices (
    id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id VARCHAR(20),
    gender VARCHAR(10),
    age INT,
    category VARCHAR(50),
    quantity INT,
    price DECIMAL(10,2),
    payment_method VARCHAR(20),
    invoice_date DATE,
    shopping_mall VARCHAR(50)
);
```

3. Select Records: Retrieve records from the table

```
SELECT * FROM Invoices;
```



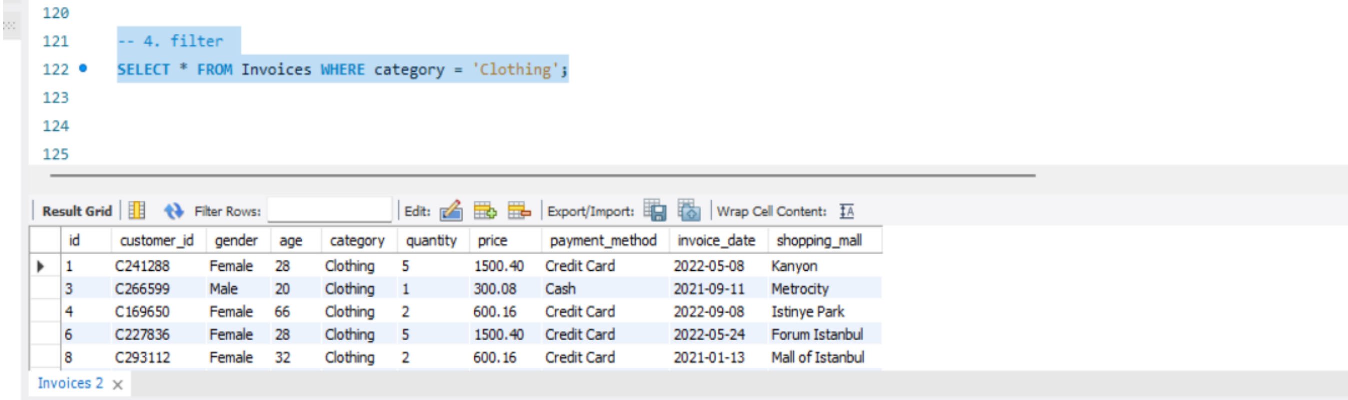
```
-- 3. select records
SELECT * FROM Invoices;
```

id	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
1	C241288	Female	28	Clothing	5	1500.40	Credit Card	2022-05-08	Kanyon
2	C111565	Male	21	Shoes	3	1800.51	Debit Card	2021-12-12	Forum Istanbul
3	C266599	Male	20	Clothing	1	300.08	Cash	2021-09-11	Metrocity
4	C169650	Female	66	Clothing	2	600.16	Credit Card	2022-09-08	Istinye Park
5	C337046	Female	53	Books	4	60.60	Cash	2021-10-24	Kanyon

4. Filter records based on certain conditions.

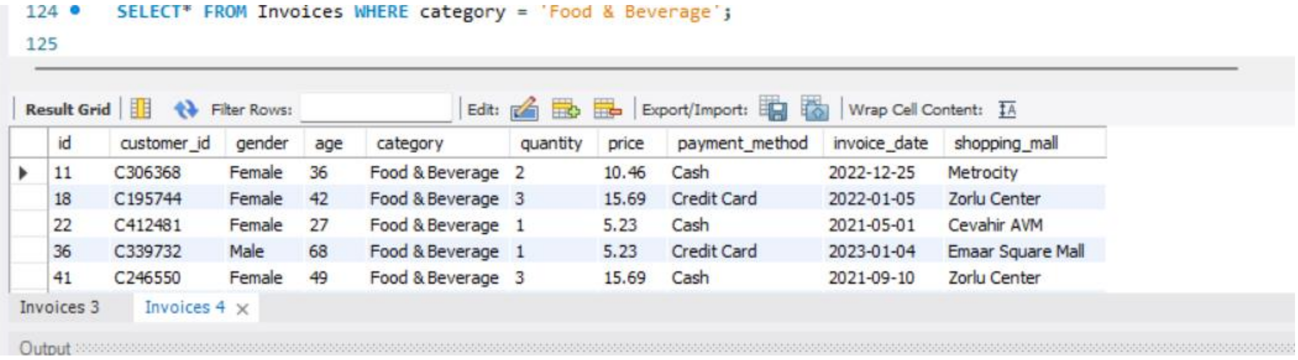
```
SELECT * FROM Invoices WHERE category = 'Clothing';

SELECT* FROM Invoices WHERE category = 'Food & Beverage';
```



```sql
-- 4. filter
SELECT * FROM Invoices WHERE category = 'Clothing';
```

id	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
1	C241288	Female	28	Clothing	5	1500.40	Credit Card	2022-05-08	Kanyon
3	C266599	Male	20	Clothing	1	300.08	Cash	2021-09-11	Metrocity
4	C169650	Female	66	Clothing	2	600.16	Credit Card	2022-09-08	Istinye Park
6	C227836	Female	28	Clothing	5	1500.40	Credit Card	2022-05-24	Forum Istanbul
8	C293112	Female	32	Clothing	2	600.16	Credit Card	2021-01-13	Mall of Istanbul

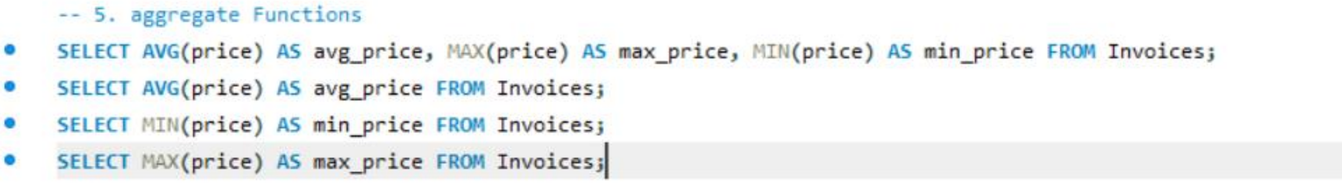


```sql
SELECT* FROM Invoices WHERE category = 'Food & Beverage';
```

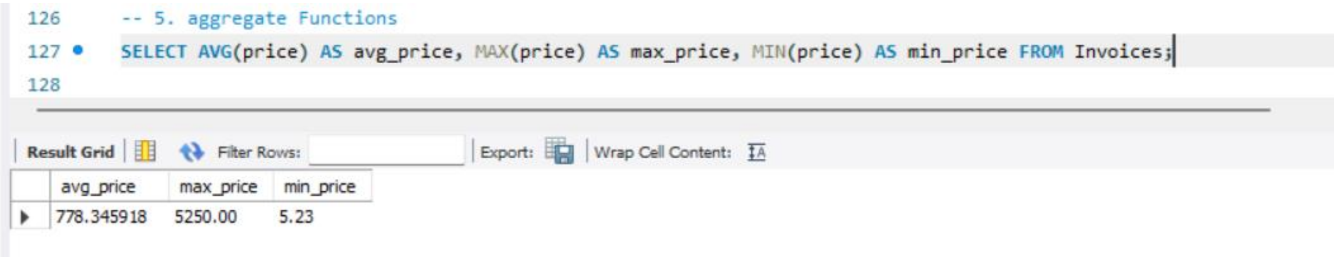
id	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
11	C306368	Female	36	Food & Beverage	2	10.46	Cash	2022-12-25	Metrocity
18	C195744	Female	42	Food & Beverage	3	15.69	Credit Card	2022-01-05	Zorlu Center
22	C412481	Female	27	Food & Beverage	1	5.23	Cash	2021-05-01	Cevahir AVM
36	C339732	Male	68	Food & Beverage	1	5.23	Credit Card	2023-01-04	Emaar Square Mall
41	C246550	Female	49	Food & Beverage	3	15.69	Cash	2021-09-10	Zorlu Center

5. Aggregate Functions: Use aggregate functions to calculate statistics

SELECT AVG(price) AS avg_price, MAX(price) AS max_price, MIN(price) AS min_price FROM Invoices;
SELECT AVG(price) AS avg_price FROM Invoices;
SELECT MIN(price) AS min_price FROM Invoices;
SELECT MAX(price) AS max_price FROM Invoices;



```sql
-- 5. aggregate Functions
SELECT AVG(price) AS avg_price, MAX(price) AS max_price, MIN(price) AS min_price FROM Invoices;
SELECT AVG(price) AS avg_price FROM Invoices;
SELECT MIN(price) AS min_price FROM Invoices;
SELECT MAX(price) AS max_price FROM Invoices;
```



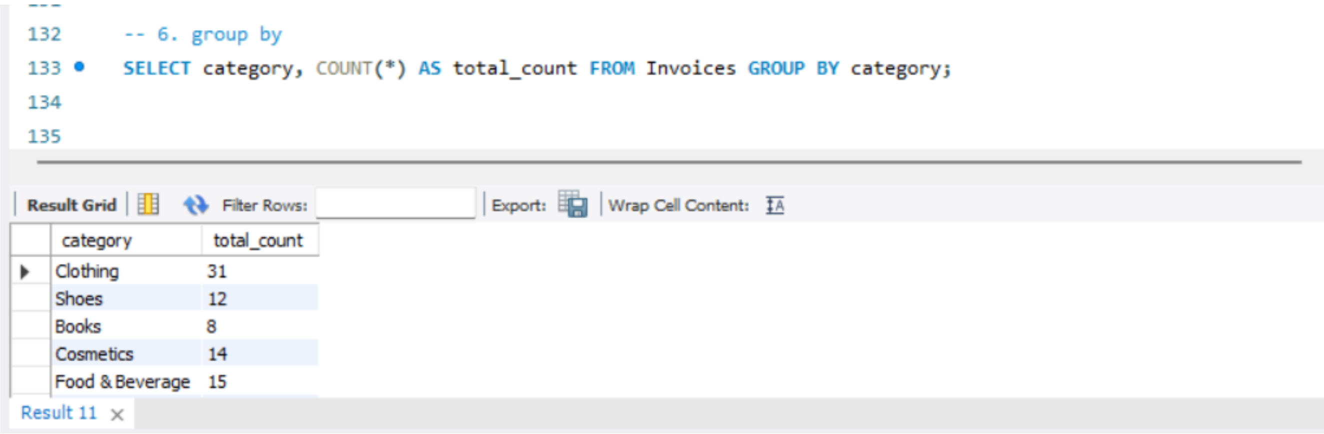
```sql
-- 5. aggregate Functions
SELECT AVG(price) AS avg_price, MAX(price) AS max_price, MIN(price) AS min_price FROM Invoices;
```

avg_price	max_price	min_price
778.345918	5250.00	5.23

6. Group By: Group data based on certain columns

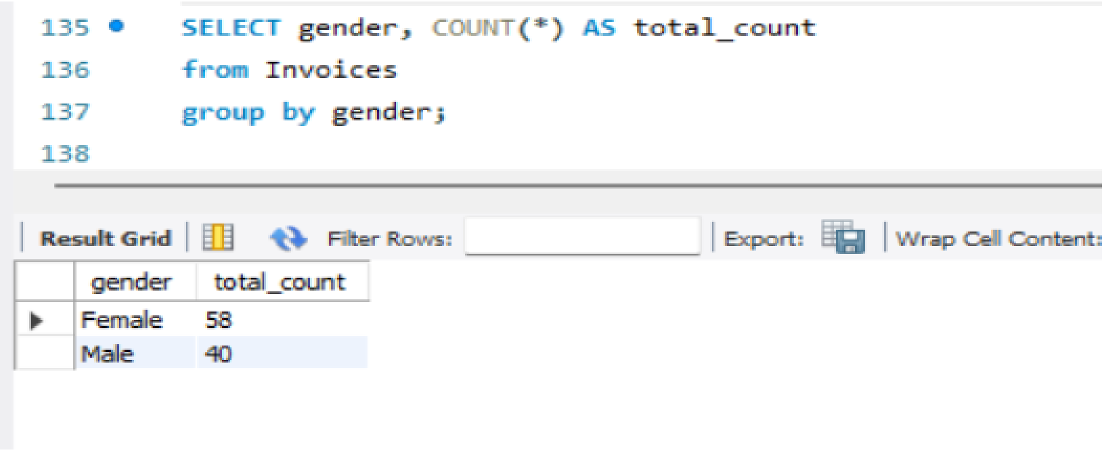
SELECT category, COUNT(*) AS total_count FROM Invoices GROUP BY category;

SELECT gender, COUNT(*) AS total_count
from Invoices
group by gender;



```
-- 6. group by
SELECT category, COUNT(*) AS total_count FROM Invoices GROUP BY category;
```

category	total_count
Clothing	31
Shoes	12
Books	8
Cosmetics	14
Food & Beverage	15

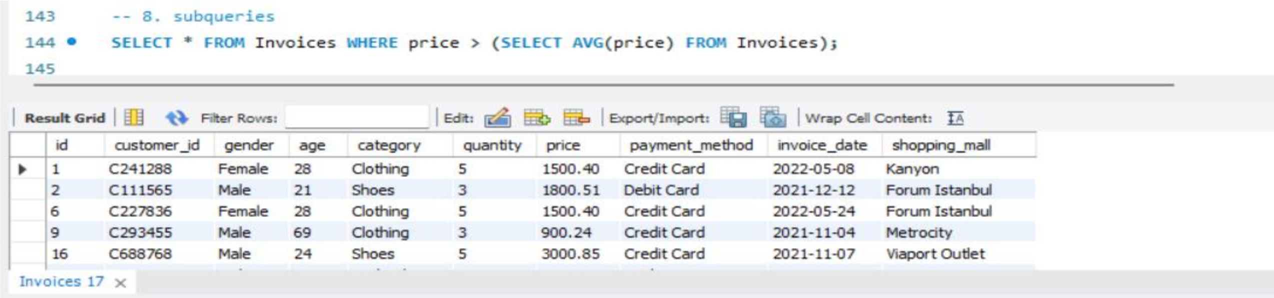


```
SELECT gender, COUNT(*) AS total_count
from Invoices
group by gender;
```

gender	total_count
Female	58
Male	40

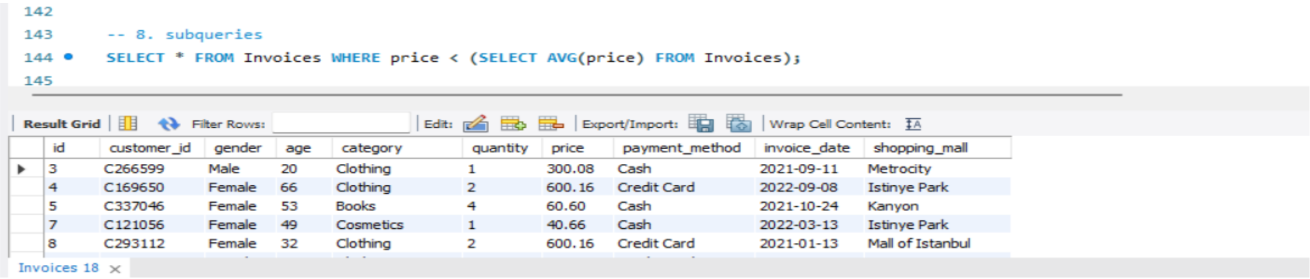
7. Subqueries: Use subqueries to retrieve data

SELECT * FROM Invoices WHERE price < (SELECT AVG(price) FROM Invoices);



```
-- 8. subqueries
SELECT * FROM Invoices WHERE price > (SELECT AVG(price) FROM Invoices);
```

id	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
1	C241288	Female	28	Clothing	5	1500.40	Credit Card	2022-05-08	Kanyon
2	C111565	Male	21	Shoes	3	1800.51	Debit Card	2021-12-12	Forum Istanbul
6	C227836	Female	28	Clothing	5	1500.40	Credit Card	2022-05-24	Forum Istanbul
9	C293455	Male	69	Clothing	3	900.24	Credit Card	2021-11-04	Metrocity
16	C688768	Male	24	Shoes	5	3000.85	Credit Card	2021-11-07	Viaport Outlet



```
-- 8. subqueries
SELECT * FROM Invoices WHERE price < (SELECT AVG(price) FROM Invoices);
```

id	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
3	C266599	Male	20	Clothing	1	300.08	Cash	2021-09-11	Metrocity
4	C169650	Female	66	Clothing	2	600.16	Credit Card	2022-09-08	Istinye Park
5	C337046	Female	53	Books	4	60.60	Cash	2021-10-24	Kanyon
7	C121056	Female	49	Cosmetics	1	40.66	Cash	2022-03-13	Istinye Park
8	C293112	Female	32	Clothing	2	600.16	Credit Card	2021-01-13	Mall of Istanbul

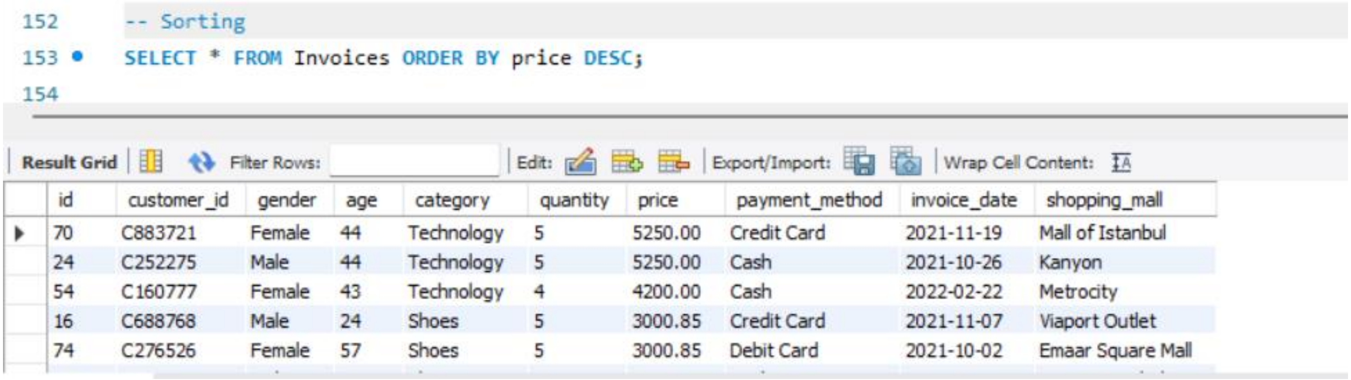
8. Update Records: Update existing records in the table

UPDATE Invoices SET price = price * 1.1 WHERE category = 'Shoes';

9. Sorting: Sort records based on certain columns

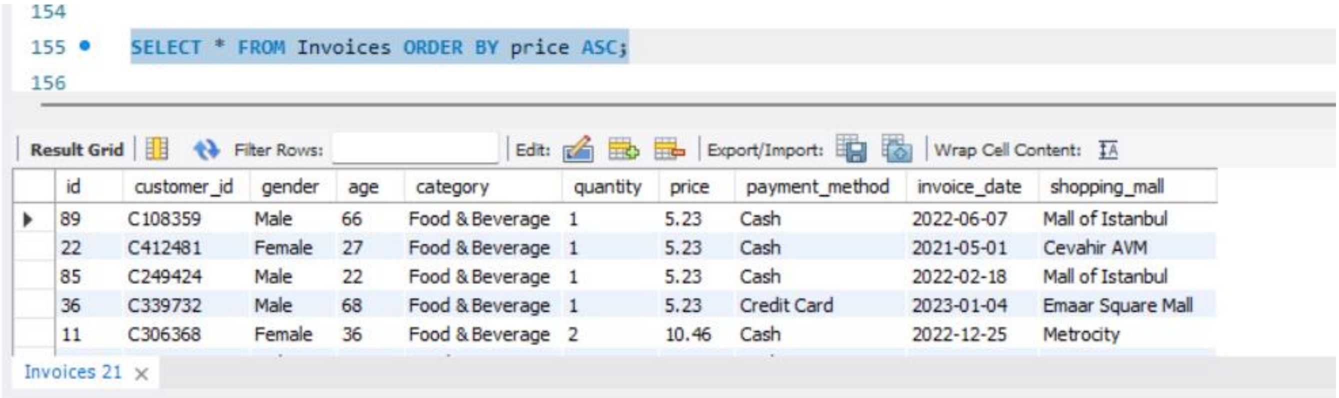
SELECT * FROM Invoices ORDER BY price DESC;

SELECT * FROM Invoices ORDER BY price ASC;



```
-- Sorting
SELECT * FROM Invoices ORDER BY price DESC;
```

id	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
70	C883721	Female	44	Technology	5	5250.00	Credit Card	2021-11-19	Mall of Istanbul
24	C252275	Male	44	Technology	5	5250.00	Cash	2021-10-26	Kanyon
54	C160777	Female	43	Technology	4	4200.00	Cash	2022-02-22	Metrocity
16	C688768	Male	24	Shoes	5	3000.85	Credit Card	2021-11-07	Viaport Outlet
74	C276526	Female	57	Shoes	5	3000.85	Debit Card	2021-10-02	Emaar Square Mall

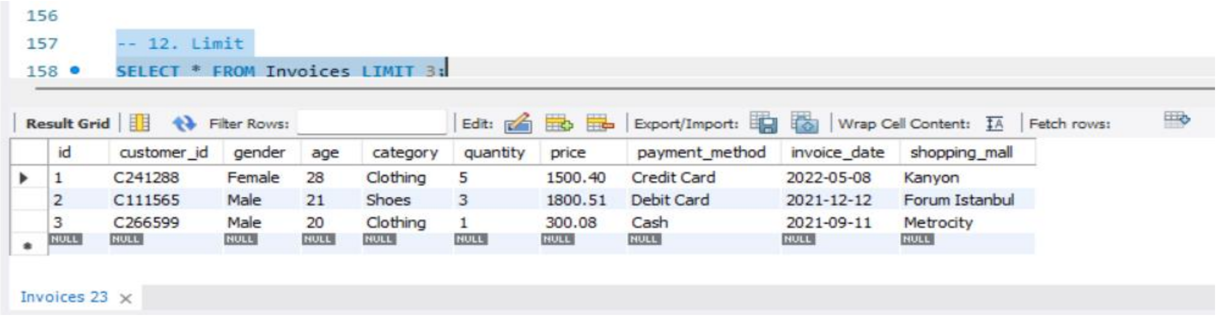


```
SELECT * FROM Invoices ORDER BY price ASC;
```

id	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
89	C108359	Male	66	Food & Beverage	1	5.23	Cash	2022-06-07	Mall of Istanbul
22	C412481	Female	27	Food & Beverage	1	5.23	Cash	2021-05-01	Cevahir AVM
85	C249424	Male	22	Food & Beverage	1	5.23	Cash	2022-02-18	Mall of Istanbul
36	C339732	Male	68	Food & Beverage	1	5.23	Credit Card	2023-01-04	Emaar Square Mall
11	C306368	Female	36	Food & Beverage	2	10.46	Cash	2022-12-25	Metrocity

10. Limit: Limit the number of records returned by a query

SELECT * FROM Invoices LIMIT 3;

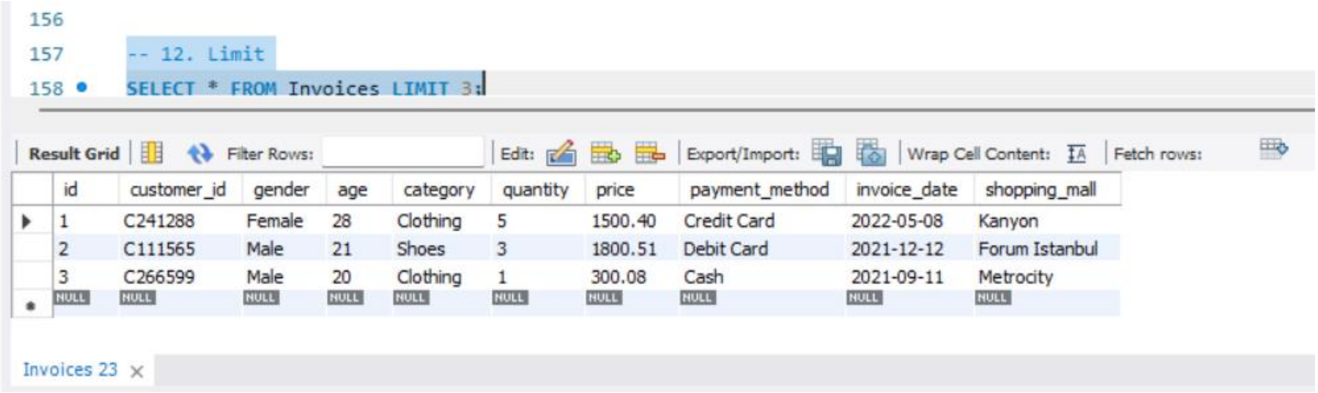


```
-- 12. Limit
SELECT * FROM Invoices LIMIT 3;
```

id	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
1	C241288	Female	28	Clothing	5	1500.40	Credit Card	2022-05-08	Kanyon
2	C111565	Male	21	Shoes	3	1800.51	Debit Card	2021-12-12	Forum Istanbul
3	C266599	Male	20	Clothing	1	300.08	Cash	2021-09-11	Metrocity
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

11. Case Statements: Use case statements for conditional operations

```
SELECT category, CASE
        WHEN price > 1000 THEN 'Expensive' ELSE 'Affordable'
    END AS price_category FROM Invoices;
```

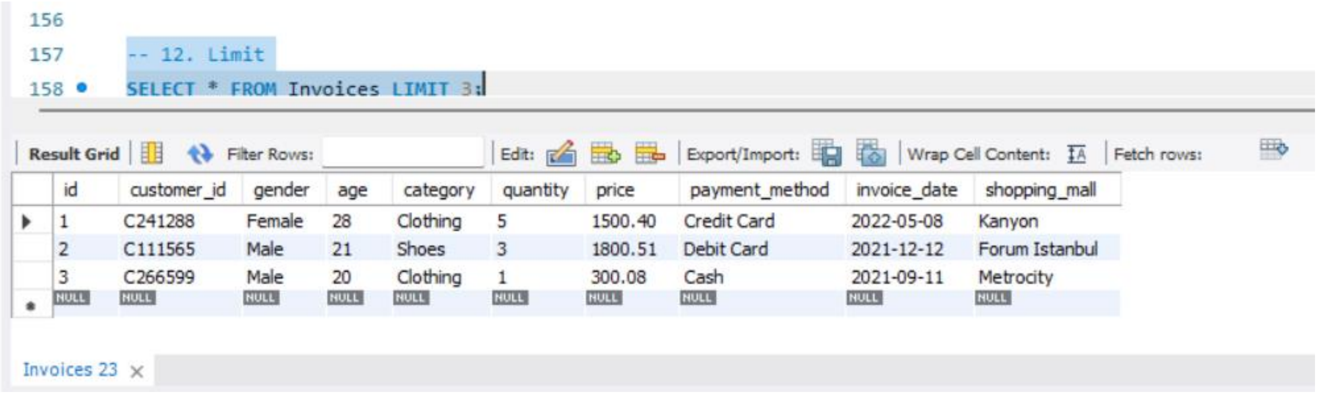


```
-- 12. Limit
SELECT * FROM Invoices LIMIT 3;
```

id	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
1	C241288	Female	28	Clothing	5	1500.40	Credit Card	2022-05-08	Kanyon
2	C111565	Male	21	Shoes	3	1800.51	Debit Card	2021-12-12	Forum Istanbul
3	C266599	Male	20	Clothing	1	300.08	Cash	2021-09-11	Metrocity
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

12. Views: Create a view based on certain criteria

```
SELECT * FROM Invoices WHERE price > 2000;
```



```
-- 12. Limit
SELECT * FROM Invoices LIMIT 3;
```

id	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
1	C241288	Female	28	Clothing	5	1500.40	Credit Card	2022-05-08	Kanyon
2	C111565	Male	21	Shoes	3	1800.51	Debit Card	2021-12-12	Forum Istanbul
3	C266599	Male	20	Clothing	1	300.08	Cash	2021-09-11	Metrocity
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Conclusion:

This SQL project provides a comprehensive overview of customer shopping analysis techniques. By leveraging SQL queries, we uncovered valuable insights into the dataset, including average purchase prices, popular product categories, and customer demographics. The project demonstrates the power of SQL in extracting meaningful information from data, which can drive informed business decisions and enhance overall performance. Whether you're a data analyst, business manager, or aspiring SQL practitioner, this project serves as a practical guide to leveraging SQL for retail analytics.